Biomedical Signal Processing and Control 86 (2023) 105186

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc

# CNN-based classification of phonocardiograms using fractal techniques

Daniel Riccio [a,b], Nadia Brancati [a,*], Giovanna Sannino [a], Laura Verde [c], Maria Frucci [a]

[a] Institute for High Performance Computing and Networking, National Research Council of Italy, Via Pietro Castellino, 111, Naples, 80131, Italy
[b] Department of Electrical Engineering and Information Technologies, University of Naples "Federico II", Via Claudio, 21, Naples, 80125, Italy
[c] Department of Mathematics and Physics, University of Studies of Campania "L. Vanvitelli", Viale Abramo Lincoln, 5, Naples, 81100, Italy

ARTICLE INFO

Dataset link: https://physionet.org, https://www.pcgfractal.icar.cnr.it

*Keywords:*
Phonocardiogram analysis
Fractals
Signal classification
CNN

ABSTRACT

Deep Learning based heart sound classification is of significant interest in reducing the burden of manual auscultation through the automated detection of signals, including abnormal heartbeats. This work presents a method for classifying phonocardiogram (PCG) signals as normal or abnormal by applying a deep Convolutional Neural Network (CNN) after transforming the signals into 2D color images. In particular, a new methodology based on fractal theory, which exploits Partitioned Iterated Function Systems (PIFS) to generate 2D color images from 1D signals is presented. PIFS have been extensively investigated in the context of image coding and indexing on account of their ability to interpolate and identify self-similar features in an image. Our classification approach has shown a high potential in terms of noise robustness and does not require any pre-processing steps or an initial segmentation of the signal, as instead happens in most of the approaches proposed in the literature. In this preliminary work, we have carried out several experiments on the database released for the 2016 Physionet Challenge, both in terms of different classification networks and different inputs to the networks, thus also evaluating the data quality. Among all experiments, we have obtained the best result of 0.85 in terms of modified Accuracy (MAcc).

## 1. Introduction

A Phonocardiogram (PCG) is the graphic display of the sound waves produced by the heart. The graphic representation of the characteristics of the sounds allows the visualizing of the temporal relationships, precise duration, intensity and contours of the waves [1]. This acquisition technique allows medical staff to register and analyze, during the auscultation of the cardiac cycle, the audible sounds and murmurs produced by the movement of the structures of the heart and the turbulence in the blood flow. During heart functioning, there are two major tones, S1 and S2, generated by the vibration of the cardiovascular system. These tones are audible during the cardiac cycle, which varies in intensity and duration. Between S1 and S2, a systolic sound is generated, principally by the closure of the atrioventricular valves. On the contrary, between S2 and S1, a diastolic sound is created by the filling of the ventricles with blood and their relaxing, see Fig. 1.

Although artificial auscultation is a convenient and low-cost cardiac diagnostic technology, physicians must have a wealth of clinical experience. There are several obstacles to the accumulation of such knowledge and training. For example, individuals may have a different auditory sensitivity. However, the distinction between different types of heart murmurs is difficult to describe [2]. Thus, over the years, many researchers have worked on the automatic classification of pathological and healthy heart sounds, but the distinction between the classes of interest is not trivial. The data is easily influenced by noise in the environment and heart sounds corresponding to different heart symptoms can be almost indistinguishable. Thus, there are still challenges that require the development of more robust methods for the early diagnosis of cardiac abnormalities.

In general, the classification task with respect to the PCG signal is performed by analyzing its features extracted in the time domain and/or frequency domain [3,4], the wavelet features [5], and/or the complexity-based features. The classification methods commonly used are Machine Learning (ML) techniques and, more recently, Deep Learning (DL) networks [6,7].

The application of ML techniques to PCG signals has resulted in a standard processing pipeline being consolidated, consisting of denoising, heartbeat segmentation, feature extraction and classification. While little research has been directed toward denoising, much attention has focused on segmentation, which is a key step. The segmentation algorithms that work best take advantage of the presence of an Electrocardiogram (ECG) signal synchronized with the PCG signal. However, this is not always available, thereby eliminating the advantage of having a much simpler hardware set-up to acquire only the PCG

* Corresponding author.
*E-mail addresses:* daniel.riccio@unina.it (D. Riccio), nadia.brancati@icar.cnr.it (N. Brancati), giovanna.sannino@icar.cnr.it (G. Sannino), laura.verde@unicampania.it (L. Verde), maria.frucci@cnr.it (M. Frucci).

https://doi.org/10.1016/j.bspc.2023.105186
Received 25 January 2023; Received in revised form 1 June 2023; Accepted 21 June 2023
Available online 1 July 2023

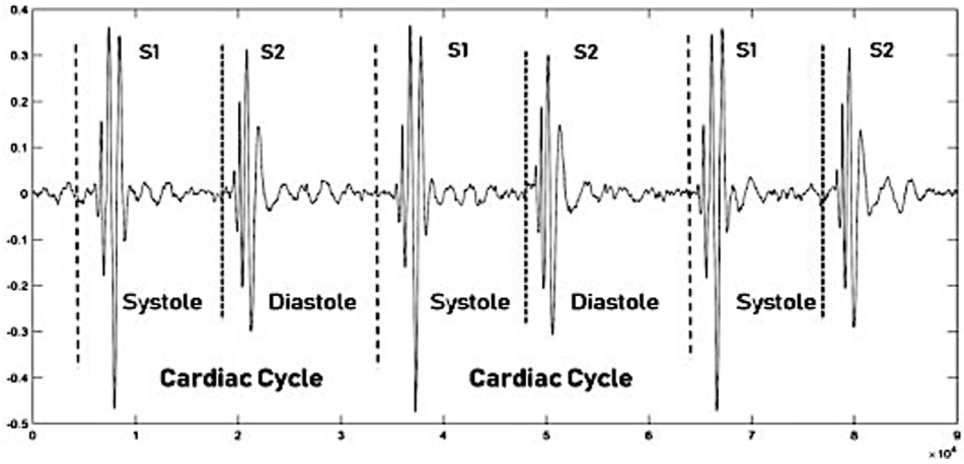


**Fig. 1.** An example of a PCG signal with the four states; S1, Systole, S2 and Diastole.

signal. Removing the need for a precise heartbeat segmentation reduces possible sources of error in the subsequent steps and imposes a greater robustness on the feature extraction process.

Indeed, one of the main contributions of the method introduced in this paper is precisely that it requires neither a denoising process nor any segmentation steps; it is able to operate directly on the input signal. As for feature extraction, a wide range of 1D handcrafted features has been proposed, among which the most widely used are based on Mel-frequency cepstral coefficients (MFCCs). Subsequently, spectral features have been introduced by windowing the signal and transforming it from 1D to 2D, so as to extract more representative features such as Discrete Wavelet Transform (DWT) coefficients. By integrating feature extraction and classification into a single end-to-end model, DL has been shown to outperform previous ML techniques. The most direct way to apply DL to PCG signals is to consider them as time series and repurpose architectures typically used to classify sequences, e.g., Long Short-Term Memory (LSTM) networks.

Accordingly, the idea of transforming a 1D signal into a 2D signal so as to exploit the greater discriminant power of Convolutional Neural Networks (Convolutional Neural Network (CNN)) models is the basis of this work. In fact, we propose a classification method based on the transformation of a 1D signal into 2D color image, which feeds a CNN network to classify the related heart sound as normal or abnormal. Indeed, one of the most significant contributions of this work is precisely the proposed transcoding technique for transforming the 1D signal into a 2D image. It has several advantages over other techniques previously used in the literature, such as DWT, both in terms of its robustness to noise and its ability to represent the features characterizing the signal.

The 1D signal to 2D color image transformation is based on the fractal theory, which exploits Partitioned Iterated Function Systems (Partitioned Iterated Function Systems (PIFS)) [8] to provide a very compact representation of a signal, capturing its salient features by considering self-similarities. Consequently, PIFS are scale-invariant, as they are able to decode a signal at resolutions other than that of the originally encoded signal. Moreover, encoding self-similarities makes them robust to noise in the input signal and allows the analysis of PCGs without any segmentation step. Unlike many other feature extraction techniques, they do not require a partitioning of the signal into beats but extract the salient features from the entire signal. The main difficulty in designing such a transcoding technique is to find a PIFS coding scheme that can reconcile its application in both 1D and 2D while maintaining compatibility between the elements of the input signal and the regions mapped in the image. This technique paves the way for interesting future research ideas, which could further improve its performance.

For the evaluation of the proposed method, we have considered the database released for the PhysioNet Computing in Cardiology Challenge 2016 (2016 PhysioNet/CinC) [9,10]. An overview of the main challenge studies, the proposed classification techniques and the corresponding results are provided in [11]. In particular, a CNN Network and a modified version of the Adaboost algorithm proposed by Potes et al. achieved the best overall score (about 0.86 of Modified Accuracy (MAcc)) in the classification of heart sounds as normal or abnormal [12]. It is important to note that all the algorithms proposed in this challenge were evaluated by using a hidden test set which has not been released by the challenge organizers. In the absence of the test set, the classification performance of the techniques proposed in the studies following the PhysioNet challenge, such as [13–15], was estimated using the only available data, namely the data contained in the training and validation sets.

Moreover, any comparison with other methods is not feasible since the other studies do not specify in detail the sample distribution among the sets. We paid particular attention to the correct use of the data without any overlap between the different sets (validation-training) and by specifying in detail as much as possible which data have been used. Aiming to allow comparisons with our method, all the data and images produced for our experiments have been made freely available at https://www.pcgfractal.icar.cnr.it.

In this preliminary work, we have carried out several experiments, both in terms of different classification networks and different inputs to the networks. Among all the experiments, we have obtained the best result of 0.85 in terms of MAcc by using a Res-Net 101.

## 2. Related works

An investigation of the literature reveals that there are several different approaches which are widely used for the classification of PCG signals based on signal processing, including ML and DL techniques. Among these, DL has certainly emerged as the most attractive solution in recent years. An exciting overview of some of the most significant related papers can be found in [6], which also presents a description of the major datasets employed in the literature, including the 2016 PhysioNet/CinC.

Although the literature on PCG signal processing, analysis and classification methods is vast [6], in this section we have decided to discuss some of those methods which are of interest in terms of the formal aspects of the proposed techniques. The existing approaches differ with respect to many characteristics such as the presence or absence of a segmentation step, the kind of features and classifier adopted for discriminating the signals into normal and abnormal, or the integration of both these aspects into the latest end-to-end models based on DL.

The presence or absence of a segmentation step is of relevant interest because it affects all subsequent processing steps. There are examples in the literature of approaches which achieve a good performance without segmenting signals, so corroborating our idea that obtaining good results even without any knowledge about the location of single heartbeat is possible. In paper [16], an unsegmented approach is presented which uses five non-linear time-scattering features based on wavelet scattering transformation from PCG recordings to classify the signal as normal or abnormal. A k-nearest neighbors (KNN) classifier with different distance functions (Euclidean, Cityblock, Chebyshev, Minkowsky, Correlation, Cosine and Spearman) has been employed to estimate the status of the heart abnormality using PCG wavelet scattering features. Li et al. [17] propose a lightweight heart sound automatic classification model in which a frequency-domain feature input feeds an improved three-layer CNN. A weighted loss function is applied to alleviate the unbalanced positive/negative portions in the samples, and all the parameters of the network are optimized.

In contrast, with respect to certain other methods an initial segmentation of individual heartbeats is mandatory, a characteristic which allows them to apply more complex feature extraction techniques. Normal et al. [18] propose a Markov-switching autoregressive process to model the raw heart sound signals directly, allowing an efficient segmentation of the cyclical heart sound states. The segmented signals were then used to train the Gaussian-mixture Hidden Markov Model classifier for the identification of abnormal beats. Chowdhury et al. [19] apply different signal processing techniques to denoise, compress and segment the PCG signals; a CNN is adopted to classify the resulting signals. First, the PCG signal is denoised and compressed using a multi-resolution analysis based on Discrete Wavelet Transformation (DWT). Then, a segmentation algorithm, based on the Shannon energy envelope and zero-crossing, is applied to segment the PCG signal into four major parts. Successively, Mel-scaled power spectrogram and MFCCs are employed to extract informative features from the PCG signal, which feeds a 5-layer feed-forward CNN.

It is important to note that all the manuscripts mentioned so far face crucial steps in the extraction of the features from the PCG signals and, only subsequently, feed these features into the classifiers. However, with the advent of DL, CNNs have become more and increasingly complex an improved performance in classification tasks. Therefore, in some recent papers, the authors have tried to avoid handcraft feature extraction. An example is provided by paper [20], where the original data are segmented by using a U-net and are classified through a multilayer CNN. In order to fully exploit the potential of CNNs, a further step has been to transform the one-dimensional signal into an image. Indeed, Ren et al. [21] propose a method based on a combination of Support Vector Machine (SVM) and VGG-16 to detect heart disorders. They first segmented the PCG files into chunks of equal length, and then, extracted a scalogram image from each chunk using a wavelet transformation. Finally, these images are used to classify the PCG signals. Similarly, in paper [22], the authors propose a classification method that analyzes the spectrograms of the signals obtained by applying short-time Fourier transformation. The generated spectrograms feed different variants of the CNN models. The transfer learning process used different datasets, apart from 2016 PhysioNet/CinC.

Although robust to noise, CNNs are potentially affected by segmentation errors of the heartbeats. Therefore, the authors of paper [23] present a 'without segmentation' approach by exploiting a Cross-wavelet transform technique and an AlexNet classifier. In detail, the heart sounds are first transformed into time–frequency spectrums along with amplitude and phase, and the obtained cross-wavelet spectrums are converted into images which feed the deep neural network.

Aiming to design a method which would be both robust to the presence of noise and versatile with respect to the inherent variability of the signal, we have investigated a new classification methodology that does not require any pre-processing and feature extraction steps or an initial segmentation. This approach is based on a signal-to-image conversion technique using fractal coding. Next, a CNN is applied to the images to classify the heart sound signals into normal and abnormal.

## 3. Methodology

In this section, we will describe in detail our proposed methodology for heart sound classification, which is illustrated in Fig. 2. First, a transcoding process transforms the 1D input signal into a 2D color image by implementing two steps. The first, namely encoding, extracts a code from the 1D input signal, while the second, called decoding, maps the code extracted in the previous step into a 2D color image.

Next, a CNN is devised to map the image to the corresponding category (normal or abnormal). We use a simple ResNet [24] as the CNN, and therefore a description of the network is not provided. The specification of the different ResNet configurations adopted in the experiments and the related setting parameters can be found in the "Experimental results" section. Here, we detail the data pre-processing dedicated to transforming a signal into an image as a result of the interpolation functions underlying fractal theory. Furthermore, the data augmentation and balancing processes involved in the experiments are also explained here.

### 3.1. Pre-processing

Before entering into the description of this phase, we consider it useful to define the context in which some of the adopted functions have been developed and the reasons that led us to choose these functions for the conversion of the signal into an image for the purposes of heart signal classification.

#### 3.1.1. Preliminaries

In 1977, Benoit B. Mandelbrot introduced fractal theory [25], in which a fractal is defined as a geometric element characterized by its non-integer dimensions and the property of reproducing the source entity at any scale. Mandelbrot argues that traditional geometry cannot represent the objects of the natural world, while fractal geometry with its non-integer dimensions provides a much more powerful tool. In 1981, John Hutchinson [26] introduced the Theory of Iterate Functions and demonstrated that there is a point of contact between classic geometry and fractal geometry. Later, Michael Barnsley, in his book *Fractals Everywhere* [27], presents the mathematical concepts behind Iterated Functions Systems (IFS), and demonstrates an important result, namely the Collage Theorem, for which an IFS can be used to represent an image. The Collage Theorem brings to light an important aspect of the related IFS as it shows that fractal theory constitutes an excellent tool for the realistic reproduction of natural entities. In the opposite sense, one might think of starting from any image and deriving an IFS that reproduces it or, at least, generates a good approximation. This problem is known as the inverse problem. In 1988, Arnaud Jacquin provided a good approximation of the solution to the inverse problem by introducing a modification of the encoding scheme provided by Barnsley, which is named Partitioned Iterated Function Systems [8]. Though originally introduced for the encoding of two-dimensional images, PIFS can be applied in order to generate a compact representation of objects at different dimensions. Indeed, they have been extended for the 3D volume compression of Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) scans. Similarly, they can be repurposed to encode lower-dimensional objects, i.e., one-dimensional signals. An interesting property of PIFS is their ability to capture the intrinsic features of the encoded object, as they rely on the representation of self-similarities. Implicitly, PIFS are able to extrapolate the distinctive features of a signal by providing a very compact descriptor, a characteristic which makes them widely used in image retrieval [28,29]. Since PIFS exploit signal self-similarities to cut out redundancies and extract salient features, we now propose a hybrid technique to encode a 1D signal using PIFS and then decode it as a 2D image in an attempt to eliminate the denoising and the signal segmentation processes.

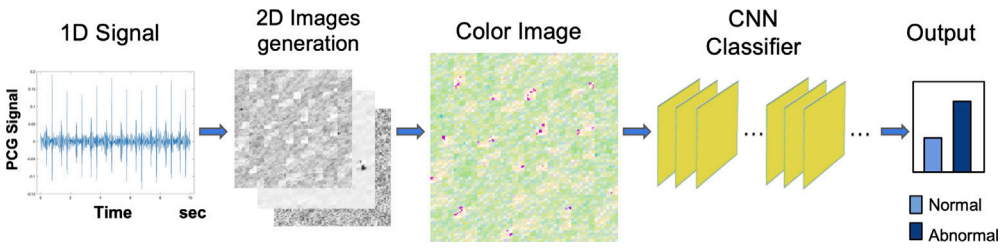


**Fig. 2.** Workflow of the method.

*3.1.2. Transcoding the 1D signal into a 2D color image*

Given a 1D signal, the method extracts three different PIFS codes so that each of them can then be decoded as a single channel of a color image. In general, PIFS encode map portions of the signal, namely domains, onto different parts of the same signal, called ranges, by means of local affine transformations, providing a compact and distinctive representation of the whole signal. Thus, the selected domain together with the parameters of the transformation represent the encoding of the range. In a second stage, a decoding process is performed, in which transformations are iteratively applied by implementing the PIFS decoding procedure introduced by Jaquin [8]. According to the Fixed Point Theorem [30], the iterative process converges producing the output 8-bit image. By keeping the range/domain association fixed, and altering the domain samples, the parameters of the transformation change, thus providing a slightly different PIFS code. In the present method, three different PIFS codes are extracted from the 1D signal. Each of these three PIFS codes produces a corresponding 8-bit image. The three images thus obtained are regarded as the three channels that combine to form the final color image which feeds a CNN classifier. Details on how the coding and decoding are performed are provided in the following section, while a scheme of the entire transcoding process is provided in Fig. 3. Moreover, the pseudocode of the encode and decode procedures is provided in Algorithms 1 and 2, respectively. Transcoding is an asymmetric process. The encoding step is more complex, namely $O(N_R \cdot log(N_R))$ (with $N_R$ the number of ranges extracted from the signal), while the decoding step is faster namely $O(N_R \cdot l_r)$ (with $l_r$ the number of samples in a range). In this case, the encoding step is applied to a 1D signal, so the complexity is greatly reduced compared to the encoding of an image.

**Algorithm 1** Encoding

**procedure** $PIFS_{code}$ = ENCODING($S_{input}$)
  $S \leftarrow$ GetFixedLength($S_{input}$)
  $R_S \leftarrow$ ExtractRanges(S)
  $D_S \leftarrow$ ExtractDomains(S)
  **for** *each* $d \in D_S$ **do**
    $d_f \leftarrow$ ComputeSaupeFeatures(d)
    insert $d_f$ in $FV_{D_S}$
  **end for**
  KDT $\leftarrow$ KDTreeBuild($FV_{D_S}$)
  **for** *each* $r \in R_S$ **do**
    $r_f \leftarrow$ ComputeSaupeFeatures(r)
    $D_{list} \leftarrow$ KDTreeQuery(KDT, $r_d$)
    $d_min \leftarrow argmin_{err}(rmse(D_{list}, r))$
    $[\alpha, \beta] \leftarrow$ approximate(r,$d_{min}$)
    insert $(d_{min}, \alpha, \beta)$ in $PIFS_{code}$
  **end for**
**end procedure**

*The encoding step*:

In the classic 2D PIFS coding scheme, an image is partitioned into a set of disjointed square regions (*ranges*). Each range is represented as the result of an affine transformation applied to another square portion of the image itself, called a *domain*. The size of the range and domain is fixed a priori. Ranges represent a coverage of the image, since each part of the image must be encoded only once. Conversely, domains may overlap, since the greater their number, the higher the probability of selecting a good domain to approximate a given range. Since there is a direct correspondence between the ranges (domains) in the 1D signal and those in the decoded image, the length of the input signal and the size of the output image are also correlated. In other words, the number of samples in the input signal must be equal to the number of pixels in the output image. Considering that the generated image will be the input to a CNN network, it is desirable to have it in a square dimension $W \times W$, which should be compatible with the input of such a network. Moreover, to facilitate the calculations, PIFS operate with ranges and domains whose size is a power of two (generally $8 \times 8$ for the ranges and $32 \times 32$ for the domains). Knowing that the ranges are to represent a coverage of the image, and following the same rationale of simplifying calculations, it is appropriate to set the size of the output image as a power of two and a multiple of the size of the range. Based on these considerations, the output image size is set as the power of two closest to the input size of the CNN network adopted as the classifier. Consequently, the number of ranges/domains depends on the size $W \times W$ of the input image. Fixing the size of both the output image and ranges automatically determines the total number of ranges to be encoded and the number of samples the input signal should consist of. Moreover, they are constant and fixed once and for all. Having all input signals of a fixed length depends on the application context, which is not the case in relation to phonocardiograms. For this reason, the variability of the PCG length must be handled appropriately before applying the encoding process to guarantee that all the input signals have a fixed length $L_{target}$.

We assume that any anomaly, if present, is located in the central part of the signal. Consequently, for a signal $S_{input}$ of $L_{input}$ length such that $L_{input} \geq L_{target}$, a new signal $S$ is obtained considering the central part of $S_{input}$ as consisting of $L_{target}$ samples. Conversely, in the case $L_{input} < L_{target}$, the signal $S$ to encode is obtained by replicating $S_{input}$ until $L_{target}$ samples can be extracted.

**Algorithm 2** Decoding

**procedure** IMAGE = DECODING($PIFS_{code}$)
  $I_h \leftarrow$ CreateMatrix(W, W, 3)
  $I_{h-1} \leftarrow I_h + err_{min}$
  **while** $||I_h - I_{h-1}||_2 \geq err_{min}$ **do**
    $I_{h-1} \leftarrow I_h$
    **for** *each* $r \in I_h$ **do**
      $[d_{min}, \alpha, \beta] \leftarrow$ GetCode($PIFS_{code}$,r)
      $d \leftarrow$ GetDomain($I_{h-1}$, $d_{min}$)
      $r \leftarrow \alpha \cdot d + \beta$
      $I_h \leftarrow$ Replace(r, $I_h$)
    **end for**
  **end while**
  IMAGE $\leftarrow I_h$
**end procedure**

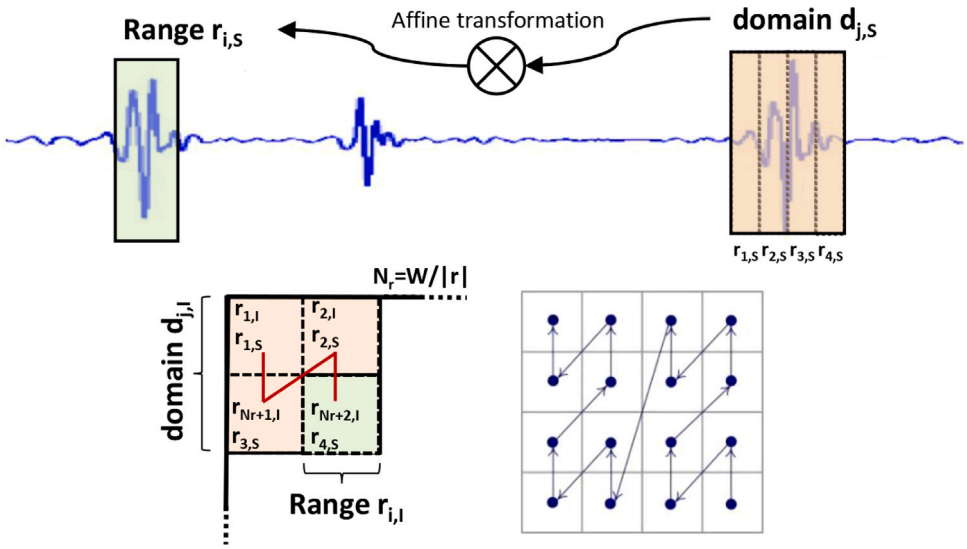


**Fig. 3.** The transcoding process.

All the signals are normalized, that is each $S$ is standardized by subtracting the mean and dividing by the standard deviation of its samples.

PIFS consist of a set of local affine contractive transformations, which exploit the signal self-similarities to cut-out redundancies, while extracting salient features. In more detail, given an input signal $S$, it is partitioned into a set $R_S = \{r_{1,S}, r_{2,S}, \dots, r_{|R_S|,S}\}$ of disjointed regions of length $|r_S|$, named *ranges*. A further set $D_S = \{d_{1,S}, d_{2,S}, \dots, d_{|D|,S}\}$ of longer regions is extracted from the same signal $S$. These regions are called *domains* and can overlap with each other. To speed up the decoding process, $|d_S|$ is usually set to $4|r_S|$. Since a range is coded by applying an affine transformation to a corresponding domain, both the ranges and domains must have the same size. Thus, the domains are extracted from a four-downsampling of $S$.

The signal $S$ is encoded range by range: for each range $r_{i,S}$, with $i = 1, \dots, |R_S|$, it is necessary to find a domain $d_{j,S}$, with $j = 1, \dots, |D_S|$, and two real numbers $\alpha$ and $\beta$ such that

$$\min_{j\in 1,\dots,|D_S|} \left\{ \min_{\alpha,\beta} \left\| r_{i,S} - (\alpha d_{j,S} + \beta) \right\|_2 \right\}. \tag{1}$$

This minimizes the quadratic error with respect to the Euclidean norm. The inner minimum in (1) on $\alpha$ and $\beta$ can be immediately computed by solving a minimum square error problem, obtaining

$$\alpha_{i,j} = \frac{\sum_{1\le k\le |r_{i,S}|}(r_{i,S}(k) - \overline{r_{i,S}})(d_{j,S}(k) - \overline{d_{j,S}})}{\sum_{1\le k\le |d_{j,S}|}(d_{j,S}(k) - \overline{d_{j,S}})^2} \tag{2}$$

$$\beta_{i,j} = \overline{r_{i,S}} - \alpha\overline{d_{j,S}}, \tag{3}$$

where $\overline{r_{i,S}}$ and $\overline{d_{j,S}}$ are the mean values of the range $r_{i,S}$ and the domain $d_{j,S}$, respectively. The outer minimum on $j$, however, requires an exhaustive search over the whole set $D$, which is very computationally expensive. Therefore, ranges and domains are classified by means of feature vectors in order to reduce the cost of searching the domain pool: if the range $r_{i,S}$ is being encoded, only the domains having a feature vector close to that of $r_i$ are considered.

In order to compute the feature vectors, Saupe's operator [31] has been adopted since it has proven to be the best compromise between discriminating power and computational simplicity. By considering a range/domain as a sequence of signal samples, Saupe's operator calculates the mean and variance of the samples. Thus, by subtracting the mean from each sample in the sequence and dividing it by the variance, Saupe's operator generates a new sequence representing the feature vector associated with the range/domain.

All the feature vectors obtained from the domains are then entered into a multidimensional search tree $T$, i.e. a KD-Tree. During the encoding process, the feature vector of each range $r_{i,S}$ is used to query the tree $T$ to select the first $n\_cd$ closest candidate domains. For each candidate domain $d_{j,S}$ the coefficients $\alpha_{i,j}$ and $\beta_{i,j}$ are computed, so that $r_{i,S} = \alpha_{i,j} * d_{j,S} + \beta_{i,j} + err_{i,j}$. Among the $n$ candidate domains, the one providing the minimum error $err_{i,j}$ is selected as the best. If the selected domain provides an $\alpha_{i,j}$ such that $|\alpha_{i,j}| > 1$ the decoding will not converge [32]. In that case, each *k-th* sample $r_{i,S}(k)$ of the range $r_{i,S}$ such that $r_{i,S}(k) > (mean(r_{i,S}) + 0.5 \cdot std(r_{i,S}))$ is set to $mean(r_{i,S})$ and $\alpha_{i,j}$ and $\beta_{i,j}$ are recomputed.

The index $j$ of the selected domain, the value of $\alpha_{i,j}$ and $\beta_{i,j}$, represents the encoding of $r_{i,S}$, while the encoding $C_S$ of the signal $S$ is given by concatenating encodes of all the ranges covering $S$.

*The decoding step*:

The decoding exploits the code computed from the 1D signal, albeit implementing the standard PIFS decoding technique of a 2D image. In other words, while in the encoding ranges $r_{i,S}$ and domains $d_{i,S}$ there are sequences of samples from a 1D signal, in the decoding these correspond to square regions of a 2D image $I$ ($r_{i,I}$ and $d_{i,I}$, respectively). Thus, the range/domain matches and $\alpha_{i,j}$ and $\beta_{i,j}$ coefficients computed from the 1D signal $S$ are now used to decode a 2D image $I$ according to the standard 2D PIFS decoding process.

For the decoding, the PIFS iteratively compute ranges $r_{i,I}$ by applying an affine transformation to the corresponding encoding domain $d_{j,I}$. The decoding process can start from any initial image as it will converge to the output image according to the Fixed Point Theorem [30].

A crucial aspect is that in 1D signals each domain includes four consecutive ranges, while in a 2D image a domain includes four adjacent ranges that form a square region that is four times the size of the range. To make the decoding process of a 2D image consistent with the code extracted from a 1D signal, the ranges $r_{k,I}$, $r_{k+1,I}$, $r_{k+2,I}$, $r_{k+3,I}$ included in the domain $d_{j,I}$ must correspond to the ranges $r_{k,S}$, $r_{k+1,S}$, $r_{k+2,S}$, $r_{k+3,S}$ included in the corresponding domain $d_{j,S}$ in the 1D signal. To maintain this correspondence, the ranges and domains are located in the image according to the Peano space-filling curve [33] which provides a z-ordering of the 2D space according to the Morton order, so that a point's position along the curve is determined by a bitwise interleaving of its coordinates [34] (see Fig. 4).

The pixel values in the decoded image $I$ are calculated by applying an iterative process described below. Let $I_h$ (with $h > 0$) be the image to be computed during the iteration $h$ and $I_0$ a completely black image. For each range $r^h_{i,I}$ the corresponding $j$ (coding domain index), $\alpha_{i,j}$ and

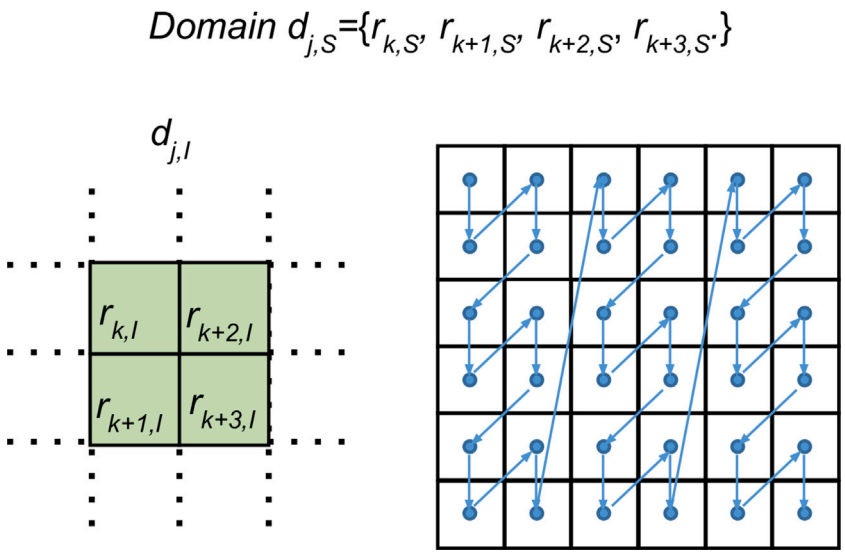


**Fig. 4.** Correspondence among ranges (domains) in 1D Signal $S$ and 2D image $I$ based on 2D Peano/Morton space filling curve.

$\beta_{i,j}$ are extracted from the encoding $C_S$. Thus, the decoding process extracts the $j$th domain from the $I_{h-1}$ image, applies a contraction (resize) by a factor of 0.25 to it to calculate $r^h_{i,I} = \alpha_{i,j} d^{h-1}_{j,I} + \beta_{i,j}$ and replaces the decoded range at the same position in the $I_h$ image. In order to ensure that the values in $r^h_{i,I}$ represent valid pixels values, a lower- and upper-bound is applied, so that $r^h_{i,I} = max(I_{min}, min(r^h_{i,I}, I_{max}))$, where $[I_{min}, I_{max}]$ represents the range of valid values for the pixels in the image. When all the ranges have been decoded and placed in $I_h$, $I_{h-1}$ is set to $I_h$ and the decoding process is repeated until $\|I_h - I_{h-1}\|_2 < err_{min}$, where $err_{min}$ is a constant value fixed a priori. The parameter $err_{min}$ estimates the convergence as a function of the difference between two consecutive iterations, and it is generally observed that $err_{min} = 10^{-3}$ is sufficient to ensure that the decoding method has converged.

The image so obtained represents the first channel of a color image. In order to increase the amount of information extracted from the signal, two further channels are generated. The second channel is obtained by a similar decoding process, where each domain is standardized according to the formula $d_{j,S} = (d_{j,S} - \overline{d_{j,S}})/std(d_{j,S})$ before the affine transformation is applied. Finally, the third channel is similarly obtained by transforming the domain according to the formula $d_{j,S} = (d_{j,S} - \overline{d_{j,S}})^2$. The color values in the resulting image $I$ are finally normalized with respect to a correction parameter $\delta$. In each of the three channels $I_c$ with $c = 1, 2, 3$, of the image $I$ the pixel values are mapped to the interval [0, $\delta$], according to the following procedure. Let $H$ be the histogram of one of the color channels of $I$. From $H$, we determine the value $k$ such that 99% of the pixels have a value within the interval $[-k, k]$. Pixels whose value lies outside that interval are saturated with respect to the nearest extreme that is $-k$ or $k$. Finally, the value of each pixel in $I_c$ is mapped to the interval $[0, \delta]$ according to the following formula:

$$I_c(i,j) = \delta \cdot log_\delta \left( \frac{max(I_c) - I_c(i,j)}{(max(I_c) - min(I_c))} \right). \quad (4)$$

## 4. Database

The data used in this study are from the database released for the PhysioNet Computing in Cardiology Challenge 2016 (2016 PhysioNet/CinC) [9,10]. It includes 3,240 heart sound signals collected from 764 adult subjects, saved in *wav* format with a length from 5 to above 120 s. The heart sound samples were recorded from different locations (aortic, pulmonic, tricuspid and mitral areas) using different acquisition sensors.

All the heart sound signals were labeled as normal or abnormal by medical experts. Precisely, the set of sound signals is constituted by 665 abnormal and 2,575 normal PCG signals and is divided into six sub-datasets (*a* through *f*). The normal sounds were acquired from healthy subjects, while the abnormal ones were collected from subjects suffering from various heart illnesses, such as heart valve defects and coronary artery diseases.

The 2016 PhysioNet/CinC database was divided into training and validation sets, while the testing set used for the challenge was blinded.

In our experiments, we considered five sub-datasets (*a* through *e*), and we used as a testing set the validation set acquired from the PhysioNet providing, of course, for the removal of the signals of the testing set from the training set.

Table 1 reports the number of abnormal and normal signals used in this study and their distribution over the training and testing sets.

In both the training and testing datasets, many signals have a low quality and, therefore, they were labeled as 'unsure' by the organizers of the Challenge [9]. The unsure signal percentages for each sub-dataset are specified in Table 2 which also summarizes the number of signals, the signal percentages and the time lengths.

Due to both the reduced number and the strong unbalancing of the data, an augmentation process was applied, which is described in the next section.

### 4.1. Data augmentation

Since the whole dataset is very unbalanced with respect to both the number of samples per class (normal and abnormal) and the origin of the signals (the sub-datasets have different numbers of total signals and each is unbalanced per class), data augmentation provides a way of balancing the dataset with respect to these two aspects. However, during the encoding step imposing a fixed length for the input signals with a length > $L_{target}$ and selecting only the central part of a signal could exclude from the image information relevant for training the classifier. To overcome this potential problem and at the same time to increase the size of the training dataset, a data augmentation process was implemented. The data augmentation process consists in taking a signal $S$ and concatenating it to itself, obtaining a signal $S_2$, which is then split into several parts, each with a length equal to $L_{target}$ and to which the transcoding process is applied. In the following paragraphs, we refer to this augmentation process as *Replication*.

Regarding the balancing of the total number of normal and abnormal signals (in the following paragraphs, we refer to this problem as

**Table 1**
Details about the heart sound signals used in this study.

| Sub-Dataset | Training set | | | Testing set | | |
|---|---|---|---|---|---|---|
| | # Abnormal signals | # Normal signals | Total signals | # Abnormal signals | # Normal signals | Total signals |
| *a* | 252 | 77 | 329 | 40 | 40 | 80 |
| *b* | 55 | 337 | 392 | 49 | 49 | 98 |
| *c* | 20 | 4 | 24 | 4 | 3 | 7 |
| *d* | 23 | 22 | 45 | 5 | 5 | 10 |
| *e* | 130 | 1905 | 2035 | 53 | 53 | 106 |
| *Total* | *480* | *2345* | *2825* | *151* | *150* | *301* |

**Table 2**
Summary of some features of 2016 PhysioNet/CinC database.

| Sub-Dataset | Tot. signals | Proportion of signals (%) | | | Signal length (s) | | |
|---|---|---|---|---|---|---|---|
| | | Abnormal | Normal | Unsure | Min | Median | Max |
| *a* | 409 | 67.5 | 28.4 | 4.2 | 9.3 | 35.6 | 36.5 |
| *b* | 490 | 14.9 | 60.2 | 24.9 | 5.3 | 8 | 8 |
| *c* | 31 | 64.5 | 22.6 | 12.9 | 9.6 | 44.4 | 122.0 |
| *d* | 55 | 47.3 | 47.3 | 5.5 | 6.6 | 12.3 | 48.5 |
| *e* | 2,141 | 6.8 | 83.2 | 9.9 | 8.1 | 21.1 | 101.7 |

*class balancing*) it is important to define the number $n_c$ of sub-signals that can be extracted from each signal $S_2$ belonging to the class with the lower number of samples. To balance two classes $A$ and $B$ with $N_A$ and $N_B$ number of signals, respectively, such that $N_A < N_B$, the value of $n_c$ is computed as $N_A/N_B$. Thus, each signal of $A$ is divided into $n_c$ sub-parts producing $n_c$ new signals. This process is applied to obtain the same number of signals per class for each sub-dataset. Regarding the process applied to obtain the same number of samples for each sub-dataset (in the following paragraphs, we refer to this problem as *dataset balancing*) the process is based on the extraction of $n_d$ sub-parts from each sample of the sub-dataset, having a number of samples $N_s < N$, where $N$ is the number of signals of the largest sub-dataset. Clearly, $n_d = N/N_s$.

A different augmentation strategy from the literature was also considered, namely the Oversample using the Adaptive Synthetic (ADASYN) algorithm [35]. The ADASYN method balances the number of samples between classes by synthetically creating new examples of the minority class through linear interpolation between the existing samples of the minority class. ADASYN extends the Synthetic Minority Oversampling Technique (SMOTE) method in that it creates more examples near the boundary between the two classes, rather than within the minority class.

## 5. Experimental results

To analyze the performance of the proposed method, several experiments were carried out, adopting different strategies for the balancing process. The results were evaluated according to the following metrics, the F1-measure, Area Under Curve (AUC) and a MAcc, which was used for the scoring of the challenge and of our experiments. MAcc is computed as an average between the Sensitivity and Specificity measures. Three different configurations of ResNet were considered separately: ResNet-18, ResNet-50 and ResNet-101. The parameter settings for both the transcoding process of a signal $S$ in an image $I$ and the training are reported in Table 3.

The results of the experiments are shown in Table 4. For the first experiment (Exp.1), no data augmentation process was applied. For the other two different experiments (Exp.2 and Exp.3) a *class balancing* was adopted by applying the ADASYN and *Replication* augmentation method, respectively. For the remaining experiments (Exp.4 and Exp.5), the *dataset balancing* was also applied, using the ADASYN and *Replication* strategy, respectively. The best measures for each experiment are highlighted in bold in Table 4 and the absolute best measures are underlined in bold. The best performance was obtained by using ResNet-101 with *dataset balancing* adopting the *Replication* augmentation process. To evaluate the leverage of each dataset on the final classification, for all the experiments, the Total Error Rate (TER) was computed:

$$TER = \frac{FP + FN}{Total\ number\ of\ samples} \tag{5}$$

where False Positive (FP) and False Negative (FN) are defined as the number of signals wrongly classified as pathological and healthy, respectively. For simplicity, in Table 5 only the results of the TER for Exp.5 are reported, but similar results were obtained in the other experiments.

The worst result was always obtained for the dataset *b*. Comparing the data in Table 2, we note that the sub-dataset *b* includes the greatest number of unsure signals, i.e. signals with a poor quality. Moreover, the median of signal lengths of the sub-dataset *b* (seconds, which corresponds to 16,000 samples at 2 KHz) is the smallest of the whole dataset. Considering that the input length $L_{target}$ required by our approach is much longer (seconds, which corresponds to 65,536 samples at 2 KHz), the signals of the sub-dataset *b* are much more affected by the replication process described in the encoding step of Section 3.1.2. These two features of the sub-dataset *b*, i.e. the high presence of unsure signals and the low median length, could explain the low performance in this sub-dataset. Therefore, further experiments were conducted according to the strategy of Exp.1 and Exp.5 and the results are reported in Table 6. In these new experiments, the sub-dataset *b* was excluded first from the training set (Exp. 1.1 and Exp. 5.1.) and later also from the testing set (Exp. 1.1 and Exp. 5.2).

Note that all the measures in terms of MAcc, F-measure and AUC are, generally, higher than those obtained with experiments including the sub-dataset *b*. In particular, the absolute best MAcc and F-measure were obtained for Exp.5.2 with a ResNet-50, while the best value of AUC was always achieved for Exp.5.2 but with a ResNet-18. As for Exp. 1.1, Resnet-18 achieved the best results in terms of MAcc, F1-measure and AUC, and was equaled by ResNet-101 only with respect to the first two.

A comparison with other methods in the literature using the Physionet/Cinc 2016 sub-datasets is not feasible because, in most cases, a detailed explanation of how the signals were partitioned into training and testing sets is not provided.

To provide a complete analysis of our results, in Table 7 we show a comparison of our approach with the methods described in Section 2, specifying the data distribution, as reported in these corresponding

**Table 3**
Parameter settings for the transcoding process and the training of the network.

| Parameters for the transcoding process | | | Parameters for the training | |
|---|---|---|---|---|
| Parameter | Description | Value | Description | Value |
| $W$ | side of I | 256 | | |
| $L_{target}$ | fixed length of S | 65536 | learning rate | 0.001 |
| $n\_cd$ | number of candidate domains | 16 | training epochs | 100 |
| $\|r_{i,S}\|$ | size of the range in S | 64 | batch size | 32 |
| $\|r_{i,I}\|$ | size of the range in I | $8 \times 8$ | network optimizer | Adam |
| $\delta$ | correction parameter | 65536 | | |

**Table 4**
Results with the different strategies of augmentation and balancing and the different ResNet configurations.

| | Balancing (Augmentation Method) | ResNet | MAcc | F1-measure | AUC |
|---|---|---|---|---|---|
| Exp.1 | NO | 18 | 0.70 | 0.64 | 0.70 |
| | | 50 | **0.71** | **0.67** | **0.71** |
| | | 101 | **0.71** | 0.65 | **0.71** |
| Exp.2 | Class Balancing (ADASYN) | 18 | 0.68 | 0.59 | **0.71** |
| | | 50 | **0.69** | **0.73** | 0.69 |
| | | 101 | 0.67 | 0.63 | 0.67 |
| Exp.3 | Class Balancing (Replication) | 18 | 0.71 | **0.71** | 0.71 |
| | | 50 | **0.74** | 0.69 | **0.74** |
| | | 101 | 0.69 | 0.64 | 0.69 |
| Exp.4 | Dataset Balancing (ADASYN) | 18 | 0.65 | **0.67** | 0.65 |
| | | 50 | **0.67** | 0.62 | **0.67** |
| | | 101 | 0.62 | 0.54 | 0.62 |
| Exp.5 | Dataset Balancing (Replication) | 18 | 0.72 | 0.71 | **0.81** |
| | | 50 | 0.73 | **0.73** | 0.73 |
| | | 101 | **0.75** | **0.73** | 0.75 |

**Table 5**
TER for each dataset starting from the results obtained with ResNet-101 in Exp.5 shown in Table 4.

| Dataset | TER |
|---|---|
| *a* | 8.31% |
| *b* | **13.95%** |
| *c* | 0.00% |
| *d* | 1.33% |
| *e* | 1.33% |

papers. We have excluded the method in [22] from the comparison since it uses further datasets.

## 6. Discussion

This is a preliminary work in which we present an architecture for mapping 1D signals into 2D color images for classification purposes. In order to stress the proposed method, we have conducted several experiments related to the classification task, evaluating different CNNs and considering different inputs for these networks.

From an analysis of Tables 4 and 6, it is clear that there is no substantial variation between the performance of the different networks in the individual experiments. This suggests that shallower networks are already able to capture the truly discriminative features present in the image. This can be attributed to the very nature of the proposed method, i.e., PIFS captures the underlying structure of the signal by concentrating it in the low frequencies, in no way favoring the noise and detail features typically carried out by the high frequencies. Such features are indeed those captured by the first layers of a CNN network and, therefore, a Resnet-18 is able to match the performance of much deeper networks such as Resnet-50 and Resnet-101.

Comparing the results of Exp.1 and Exp.5 (shown in Table 4) with those reported in Table 6, there is an appreciable variation in performance, when *b* was eliminated from both the training and testing sets (Exp.1.2 and Exp.5.2). This suggests, that *b* is indeed representative of a strong variability. However, when sub-dataset *b* was eliminated from the training set only (Exp.1.1 and Exp.5.1), the change in performance is very slight. Given that it consists of more than a few samples, it can be inferred that the method is particularly robust, as well as a good generalization ability. Moreover, considering that it does not benefit from this information in training, it is evident that it is still able to maintain a good performance.

The proposed balancing method produces a significant performance increase unlike ADASYN, whose application worsens the performance, albeit live, also with respect to the method without any augmentation. This is attributable to the fact that ADASYN tries to estimate the distribution of input signals for the purpose of synthesizing new ones as a combination of neighboring elements in the input space. Given the size of the input space and the limited number of signals available, the synthesized ones are likely to be subject to, which introduce false anomalies into the generated signals. In contrast, the introduced balancing technique is based on the replication of real sequences of samples in the input, so it strongly limits the potential introduction of artifacts to only the junction points of the replications. Moreover, balancing at the sub-dataset level is more effective than its application at the class level. This can be explained in terms of a consideration that the signals in each dataset have inherent characteristics arising from the context and the devices with which they were acquired. Dataset balancing provides a sufficient number of signals to represent each of these conditions adequately. As regards the comparisons with the methods shown in Table 7, such an analysis is prevented by the fact that no knowledge is provided of how the signals from the Physionet/Cinc 2016 dataset were organized for the experiments. Nevertheless, we can highlight the good performance of our method.

PIFS have been extensively explored in image indexing and retrieval due to their ability to capture structural information in an image while leaving out details and noise. For this reason, PIFS have been considered as a central element of our transcoding process, which is robust to noise and does not require any pre-processing or segmentation of the signal for exactly this reason. This last aspect is of paramount

**Table 6**
Results of the experiments Exp.1 and Exp.5 excluding dataset *b* from the training set (Exp.1.1, Exp.5.1) and also from the testing set (Exp.1.2, Exp.5.2).

| | Balancing (Augmentation Method) | ResNet | MAcc | F1-measure | AUC |
|---|---|---|---|---|---|
| Exp.1.1 | NO | **18** | **0.70** | 0.63 | **0.71** |
| | | **50** | **0.70** | 0.65 | 0.70 |
| | | **101** | **0.70** | **0.69** | 0.70 |
| Exp.1.2 | NO | **18** | **0.79** | **0.79** | **0.81** |
| | | **50** | 0.78 | 0.77 | 0.78 |
| | | **101** | **0.79** | **0.79** | 0.79 |
| Exp.5.1 | Dataset Balancing (Replication) | **18** | 0.74 | **0.75** | **0.81** |
| | | **50** | **0.75** | 0.72 | 0.75 |
| | | **101** | 0.73 | 0.71 | 0.73 |
| Exp.5.2 | Dataset Balancing (Replication) | **18** | 0.84 | 0.84 | **0.91** |
| | | **50** | **0.85** | **0.86** | 0.85 |
| | | **101** | 0.84 | 0.85 | 0.84 |

**Table 7**
The results of other methods that have worked on the 2016 Physionet/CinC dataset.

| Year | Ref | MAcc | Distribution data |
|---|---|---|---|
| 2022 | [16] | 0.97 | training 70%, testing 30% |
| 2021 | [17] | 0.86 | training 60%, validation 20%, testing 20% |
| 2019 | [18] | 0.86 | Training 2072, testing 800 |
| 2021 | [20] | 0.87 | not specified |
| 2020 | [19] | 0.97 | training 90%, validation 10% |
| 2018 | [21] | 0.56 | 3-fold cross-validation on e, b, and f sub-datasets |
| 2021 | [23] | 0.98 | Training 3240, testing 301 |

importance, as it makes our method extremely extensible and applicable to signals of a very different nature without requiring any special adjustments or modifications.

Another important advantage of the proposed method is its potential application with lighter networks on lower-performing devices. Indeed, PIFS are asymmetric schemes in which the encoding is computationally expensive,but the decoding particularly fast. In the specific case, the encoding is applied to a 1D signal, so it is much less expensive than if applied to a 2D image, while only the light part of the transcoding process, i.e., the decoding, involves a 2D signal, making the whole method high-performing and particularly fit for purpose. Moreover, PIFS are interpolators, so the same code could be decoded at different resolutions (higher or lower). This may be an advantage when networks are able to process input images at higher resolutions. At the same time, the code could be decoded at lower resolutions, without going through any image resizing which would result in a greater loss of information, working with lighter networks on lower-performing devices.

As this is an initial work in the use of PIFS as a signal transcoding tool, this method demonstrates the feasibility of the process, but leaves room for improvements. First, further research is needed to overcome the fixed length limitation of the input signal. In a real application, the system could be set to acquire the signal of the desired length. However, if it is necessary to process pre-existing signals or signals acquired from other applications, the limitation on length persists. As detailed in Section 3.1.2, the current method truncates signals that are too long at the central part and replicates signals that are too short by spliced repetitions. This problem is the main issue to be investigated in future extensions of the method.

Another aspect worthy of attention is the way the different color channels are generated. Currently, domain transformation provides a different representation of the corresponding range in each channel, thus increasing the information content of the whole color image. This choice has been found to be the best performing method downstream of a range of possible options, corroborated by the fact that it is consistent with the approach which has been implemented in other works adopting PIFS for image indexing purposes.

However, this finding does not eliminate the strong correlation that there can be among the three channels of the image derived from a signal. A possible line of investigation could be to maintain the same range/domain relationship for the different channels, while changing the representation domain, i.e., spatial for the first channel and spectral for the remaining two, based on different transforms, such as the Fast Fourier Transform and Discrete Wavelet Transform.

## 7. Conclusions and future works

This paper proposes a new fractal-based method of 1D to 2D signal transcoding, which maps the input signal to a color image in order to benefit from CNNs which are performing increasingly impressively in classification tasks. The contributions of this work are several. First, it is an initial work on the possibility of using the PIFS coding scheme for signal transcoding. Indeed, one of the main difficulties has been precisely that of readjusting the original definition of the encoding scheme so that it is compatible with 1D encoding and 2D decoding. Secondly, the choice of using PIFS has been dictated by their ability to operate directly on the input signal without requiring any denoising or segmentation steps. This aspect makes the method particularly attractive because of its robustness to noise and signal variability. Moreover, PIFS are able to extract features which are captured by the first layers of a CNN network and, therefore, a Resnet-18 is able to match the performance of much deeper networks. This characteristic makes our approach particularly suitable for implementation on devices with a low computational power.

However, the current definition of the transcoding scheme imposes a fixed length on the input signal, making a truncation or replication operation necessary if the signal is longer or shorter than the predetermined length. Our future work will address the removal of this limitation by considering the development of an encoding process that generates images whose size depends on the number of samples of the PCG signals and the use of a classifier that can receive input images which have a variable resolution. Therefore, new experiments will be carried out in the near future regarding the transcoding processes, classification networks and different databases.

## CRediT authorship contribution statement

**Daniel Riccio:** Conceptualization, Methodology, Software, Validation, Writing – original draft, Writing – review and editing, Supervision. **Nadia Brancati:** Conceptualization, Methodology, Validation, Writing – original draft, Writing – review and editing. **Giovanna Sannino:** Validation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Laura Verde:** Software, Validation, Resources, Data curation, Writing – original draft, Writing – review

& editing, Visualization. **Maria Frucci:** Conceptualization, Methodology, Validation, Writing – original draft, Writing – review & editing, Supervision.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Co-author currently employed as associate editor by BSPC journal - Giovanna Sannino.

## Data availability

The 2016 PhysioNet/CinC challenge dataset is available at https://physionet.org and all the data and images produced for our experiments are available at https://www.pcgfractal.icar.cnr.it.

## Acknowledgment

All authors have read and agreed to the published version of the manuscript.

## References

[1] C. Ahlström, Processing of the Phonocardiographic Signal: Methods for the Intelligent Stethoscope (Ph.D. thesis), Institutionen för medicinsk teknik, 2006.
[2] J. Martinez-Alajarin, R. Ruiz-Merino, Efficient method for events detection in phonocardiographic signals, in: Bioengineered and Bioinspired Systems II, Vol. 5839, International Society for Optics and Photonics, 2005, pp. 398–409.
[3] P.T. Krishnan, P. Balasubramanian, S. Umapathy, Automated heart sound classification system from unsegmented phonocardiogram (PCG) using deep neural network, Phys. Eng. Sci. Med. (2020) 1–11.
[4] T. Koike, K. Qian, Q. Kong, M.D. Plumbley, B.W. Schuller, Y. Yamamoto, Audio for audio is better? An investigation on transfer learning models for heart sound classification, in: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society, EMBC, IEEE, 2020, pp. 74–77.
[5] M. Baydoun, L. Safatly, H. Ghaziri, A. El Hajj, Analysis of heart sound anomalies using ensemble learning, Biomed. Signal Process. Control 62 (2020) 102019.
[6] S. Ismail, B. Ismail, I. Siddiqi, U. Akram, PCG classification through spectrogram using transfer learning, Biomed. Signal Process. Control 79 (2023) 104075.
[7] W. Chen, Q. Sun, X. Chen, G. Xie, H. Wu, C. Xu, Deep learning methods for heart sounds classification: A systematic review, Entropy 23 (6) (2021) 667.
[8] A.E. Jacquin, A Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding, Georgia Institute of Technology, 1989.
[9] C. Liu, D. Springer, Q. Li, B. Moody, R.A. Juan, F.J. Chorro, F. Castells, J.M. Roig, I. Silva, A.E. Johnson, et al., An open access database for the evaluation of heart sound algorithms, Physiol. Meas. 37 (12) (2016) 2181.
[10] A.L. Goldberger, L.A. Amaral, L. Glass, J.M. Hausdorff, P.C. Ivanov, R.G. Mark, J.E. Mietus, G.B. Moody, C.-K. Peng, H.E. Stanley, PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals, Circulation 101 (23) (2000) e215–e220.
[11] G.D. Clifford, C. Liu, B. Moody, D. Springer, I. Silva, Q. Li, R.G. Mark, Classification of normal/abnormal heart sound recordings: The PhysioNet/Computing in cardiology challenge 2016, in: 2016 Computing in Cardiology Conference, CinC, IEEE, 2016, pp. 609–612.
[12] C. Potes, S. Parvaneh, A. Rahman, B. Conroy, Ensemble of feature-based and deep learning-based classifiers for detection of abnormal heart sounds, in: 2016 Computing in Cardiology Conference, CinC, IEEE, 2016, pp. 621–624.
[13] F. Noman, C.-M. Ting, S.-H. Salleh, H. Ombao, Short-segment heart sound classification using an ensemble of deep convolutional neural networks, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2019, pp. 1318–1322.
[14] S. Latif, M. Usman, R. Rana, J. Qadir, Phonocardiographic sensing using deep learning for abnormal heartbeat detection, IEEE Sens. J. 18 (22) (2018) 9393–9400.
[15] J.P. Dominguez-Morales, A.F. Jimenez-Fernandez, M.J. Dominguez-Morales, G. Jimenez-Moreno, Deep neural networks for the recognition and classification of heart murmurs using neuromorphic auditory sensors, IEEE Trans. Biomed. Circuits Syst. 12 (1) (2017) 24–34.
[16] S. Ajitkumar Singh, N. Dinita Devi, S. Majumder, An improved unsegmented phonocardiogram classification using nonlinear time scattering features, Comput. J. (2022).
[17] T. Li, Y. Yin, K. Ma, S. Zhang, M. Liu, Lightweight end-to-end neural network model for automatic heart sound classification, Information 12 (2) (2021) 54.
[18] F. Noman, S.-H. Salleh, C.-M. Ting, S.B. Samdin, H. Ombao, H. Hussain, A Markov-switching model approach to heart sound segmentation and classification, IEEE J. Biomed. Health Inf. 24 (3) (2019) 705–716.
[19] T.H. Chowdhury, K.N. Poudel, Y. Hu, Time-frequency analysis, denoising, compression, segmentation, and classification of PCG signals, IEEE Access 8 (2020) 160882–160890.
[20] Y. He, W. Li, W. Zhang, S. Zhang, X. Pi, H. Liu, Research on segmentation and classification of heart sound signals based on deep learning, Appl. Sci. 11 (2) (2021) 651.
[21] Z. Ren, N. Cummins, V. Pandit, J. Han, K. Qian, B. Schuller, Learning image-based representations for heart sound classification, in: Proceedings of the 2018 International Conference on Digital Health, 2018, pp. 143–147.
[22] K.N. Khan, F.A. Khan, A. Abid, T. Olmez, Z. Dokur, A. Khandakar, M.E. Chowdhury, M.S. Khan, Deep learning based classification of unsegmented phonocardiogram spectrograms leveraging transfer learning, Physiol. Meas. 42 (9) (2021) 095003.
[23] P. Dhar, S. Dutta, V. Mukherjee, Cross-wavelet assisted convolution neural network (AlexNet) approach for phonocardiogram signals classification, Biomed. Signal Process. Control 63 (2021) 102142.
[24] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[25] B.B. Mandelbrot, Fractals, Vol. 1, WH freeman New York, 1977.
[26] J.E. Hutchinson, Fractals and self similarity, Indiana Univ. Math. J. 30 (5) (1981) 713–747.
[27] M.F. Barnsley, Fractals Everywhere, Academic Press, 2014.
[28] R. Distasi, M. Nappi, M. Tucci, FIRE: Fractal indexing with robust extensions for image databases, IEEE Trans. Image Process. 12 (3) (2003) 373–384.
[29] M. De Marsico, M. Nappi, D. Riccio, FARO: Face recognition against occlusions and expression variations, IEEE Trans. Syst., Man, Cybern.-Part A: Syst. Hum. 40 (1) (2009) 121–132.
[30] R.F. Brown, Fixed Point Theory and Its Applications: Proceedings of a Conference Held at the International Congress of Mathematicians, August 4-6, 1986, Vol. 72, American Mathematical Soc., 1988.
[31] D. Saupe, Accelerating fractal image compression by multi-dimensional nearest neighbor search, in: Proceedings DCC'95 Data Compression Conference, IEEE, 1995, pp. 222–231.
[32] A.E. Jacquin, et al., Image coding based on a fractal theory of iterated contractive image transformations, IEEE Trans. Image Process. 1 (1) (1992) 18–30.
[33] H. Sagan, Space-Filling Curves, Springer Science & Business Media, 2012.
[34] G.M. Morton, A Computer Oriented Geodetic Data Base and a New Technique in File Sequencing, International Business Machines Company, 1966.
[35] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.